Visión por Computadora 2024

Lab 3

30.abril.2024

En este lab vamos a implementar una U-net para filtrar imágenes en escala de grises.

1. Investigar sobre los filtros anisotrópicos, en qué se diferencian de un filtro como el *gaussian blur*, y cómo funcionan. Principalmente usaremos el filtro anisotrópico de Perona-Malik.

   Descargar el archivo `Anisotropic.py` en el cual ya están implementadas dos funciones: `anisodiff` y `anisodiff3`, las cuales sirven para aplicar un filtro anisotrópico sobre imágenes en escala de grises, e imágenes en format RGB, respectivamente.

2. Descargar la base de datos BSD500, de Berkeley, la cual originalmente es una base de datos para evaluar algoritmos de segmentación. En esta ocasión únicamente usaremos la base para generar imágenes a las que aplicaremos filtros anisotrópicos.

   Una réplica de esta base de datos puede encontrarse en: `https://github.com/BIDS/BSDS500`.
   Trabjar únicamente con las imágenes de Test en: `SDS500/BSDS500/data/images/test/`.

3. Convertir las imágenes del dataset a escala de grises, y luego, a cada una, aplicar el filtro anisotrópico. Se sugiere utilizar los parámetros:

   `niter=50, kappa=20, gamma=0.2, step=(1.,1.), option=1, ploton=False`.

4. De la colección de pares (I_i, F_i) (imagen original, imagen filtrada), producir una colección de muestras $(\mathbf{x}_i, \mathbf{y}_i)$, donde

   $$\mathbf{x}_i = \text{ventana de tamaño } k \times k, \text{ en la imagen original} \mathbf{y}_i = \text{ventana de tamaño } k \times k, \text{ en la imagen filtrada}$$

   ambas ventanas deben corresponder a la misma región (salvo que $\mathbf{x}_i$ viene de la imagen original I, mientras que la $\mathbf{y}_i$ es la misma región pero tomada de la imagen filtrada.
   La posición de la ventana puede ser tomada de forma aleatoria siempre que se encuentre dentro de la imagen. El tamaño $k \times k$ de las ventanas es un parámetro que ustedes deben decidir.
   Sugerencia: usar potencias de 2, por ejemplo: 16×16, 32×32 ó 64×64.

   Construir una base de datos de pares $(\mathbf{x}_i, \mathbf{y}_i)$ de ventanas. Considere un número bastante amplio de estas ventanas. E.g. $N = 5 \times 10^5$ o más.
   Separar estas ventanas en un conjunto de entrenamiento, otro de validación, y otro de prueba.

5. Entrenar una red neuronal U-net (con 3 ó 4 niveles de profundidad), con los pares de ventanas $(\mathbf{x}_i, \mathbf{y}_i)$ en el conjunto de entrenamiento. Calibrar los parámetros y refinar en entrenamiento de su red neuronal hasta que ustedes se sientan confiados en el desempeño de su red.

   La red debe estar diseñada para recibir un stack de ventanas, o tensor de tamaño $(?, k, k, 1)$, y la salida debe ser también un stack o tensor de ventanas de tamaño $(?, k, k, 1)$.

6. La inferencia: Para hacer la inferencia sobre una imagen de test debe hacer un barrido de ventanas que cubra su imagen. El tamaño de estas ventanas debe coincidir con el tamaño $k \times k$ usando en el entrenamiento de la U-net.

   Con estas ventanas, usted construirá un tensor $(?, k, k, 1)$ que alimentará a la red neuronal. Posterior a la inferencia, la salida es de nuevo un tensor $(?, k, k, 1)$ que corresponde a la colección de ventanas ya filtradas con el filtro anisotrópico.

   Con este tensor de salida, deberá reconstruir la imagen filtrada, colocando cada ventana en su posición correspondiente de donde fue tomada. Los traslapes se deben resolver promediando los resultados de las diferentes ventanas que coincidan sobre un píxel.